**2004 Proc. Int. Conf. on Machine Automation,
pp. 453-456, Osaka, Japan, Nov. 2004.**

Teaching a Mobile Robot to Take Elevators

Koji Iwase, Jun Miura, and Yoshiaki Shirai
Dept. of Computer-Controlled Mechanical Systems,
Osaka University, Suita, Osaka 565-0871, Japan
{iwase,jun,shirai}@cv.mech.eng.osaka-u.ac.jp

Abstract

The ability of moving between floors by using elevators is indispensable for mobile robots operating in office environments to expand their work areas. This paper describes a method of interactively teaching the task of taking elevators for making it easier for the user to use such robots for various elevators. The necessary knowledge of the task is organized as the task model*. The robot examines the task model and determines what are missing in the model, and then asks the user to teach them. This enables the user to teach the necessary knowledge easily and efficiently. Experimental results show the potential usefulness of our approach.*

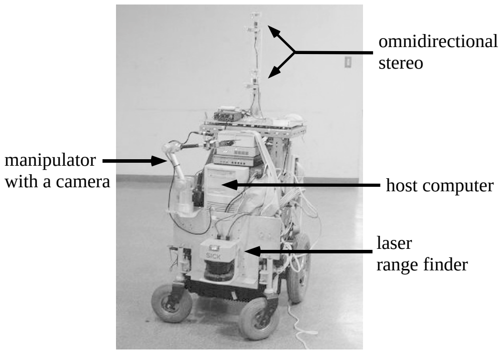


Fig. 1: Our mobile robot.

1 Introduction

The ability of moving between floors by using elevators is indispensable for mobile robots performing service tasks in office environments to extend their working areas. We have developed a mobile robot that can take elevators, but we had to give the robot in advance the necessary knowledge such as the shape of the elevator and the positions of the buttons. Since the necessary knowledge of the task of taking elevators is different from place to place, it is desirable that the user can easily teach such knowledge on-site.

We have been developing a teaching framework called *task model-based interactive teaching* [9], in which the robot examines the description of a task, called *task model*, to determine missing pieces of necessary knowledge, and actively asks the user to teach them. We apply this framework to the task of taking elevators (*take-an-elevator* task) by our robot (see Fig. 1). This paper describes the task models and the interactive teaching method with several teaching examples.

2 Task Model-Based Interactive Teaching

Interaction between the user and a robot is useful for an efficient and easy teaching of task knowledge. Without interaction, the user has to think by himself/herself about what to teach to the robot. This is difficult for the user partly because he/she does not have enough knowledge of the robot's ability (i.e., what the robot can (or cannot) do), and partly because the user's knowledge may not be well-structured. If the robot knows of *what are needed* for achieving the task, then the robot can ask the user to teach them; this enables the user to easily give necessary knowledge to the robot. This section explains the representations for task models and the teaching strategy.

2.1 Task Model

In our interactive teaching framework, the knowledge of a task is organized in a *task model*, in which necessary pieces of knowledge and their relationships are described. Some pieces of knowledge require other ones; for example, a procedure for detecting an object may need the shape or the color of the object. Such dependencies are represented by the network of knowledge pieces. The robot examines what are given and what are missing in the task model, and asks the user to teach the missing pieces of knowledge.

Hierarchical Task Structure Robotic tasks usually have hierarchical structures. Fig. 2 shows a hierarchy of robot motions for the *take-an-elevator* task. For example, a subtask, *move and push button*, is further decomposed into two steps (see the bottom of the figure): moving to the position where the robot can push the button, and actually pushing the button by the manipulator using visual feedback. Such a hierarchical task structure is the most basic representation in the task model.

Non-terminal nodes in a hierarchical task structure are *macros*, which are further decomposed into more specific subtasks. Terminal nodes are *primitives*, the achievement of which requires actual robot motion and sensing operations.

Robot and Object Models The robot model describes knowledge of the robot system such as the size and the mechanism of components (e.g., a mobile base and an arm) and the function and the position of sensors (e.g., cameras and range finders). Object models describe object properties including geometric ones, such as size, shape, and pose, and photometric ones related to visual recognition.

Movements The robot has two types of movements: free movement and guarded movement. A free movement is the one that the robot is required to a given destination without colliding with obstacles; the robot does not need to follow a specific trajectory. On the other hand, in a guarded move-

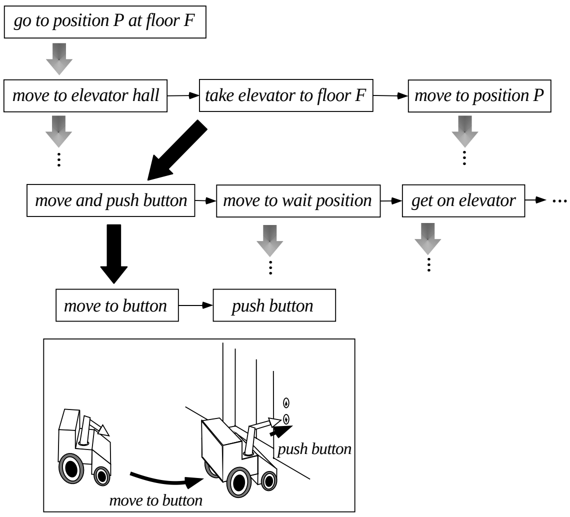
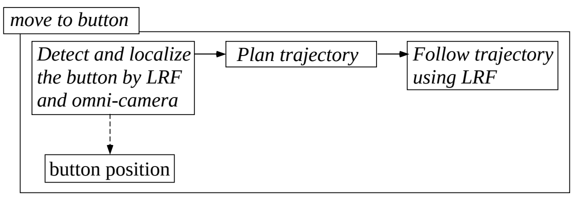
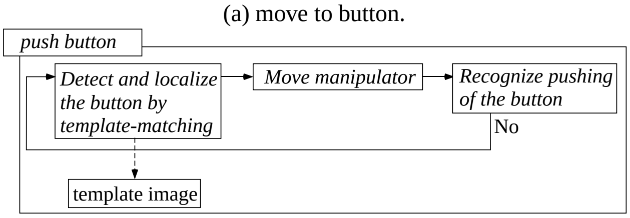


Fig. 2: A hierarchical structure of the *take-an-elevator* task.



(a) move to button.



(b) push the button.

Fig. 3: Diagrams for example primitives. Dashed lines indicate dependencies.

ment, the robot has to follow some trajectory, which is usually generated from the configuration of surrounding obstacles; movements of this type are basically used for reaching a specific pose (position and orientation) or for passing through a narrow space. Fig. 3(a) shows the diagram for the subtask of moving to the position where the robot can push a button.

Hand Motions Hand motions are described by its trajectory. They are usually implemented as sensor-feedback motions. Fig. 3(b) shows the diagram for the subtask of pushing a button.

Sensing Skills A sensing operation is represented by a *sensing skill*. Sensing skills are used in various situations such as detecting and recognizing objects, measuring properties of objects, and verifying conditions on the geometric relationship between the robot and the objects.

2.2 Interactive Teaching Using Task Model

The robot tries to perform a task in the same way even in the case where some pieces of knowledge are missing. When the robot cannot execute a motion because of a missing piece of knowledge, the robot pauses and generates a query to the user for obtaining it. By repeating this process, the robot completes the task model with leading the interaction with the user. It could be possible to examine the whole task model before execution and to generate a set of queries for missing pieces of knowledge.

3 Analysis of Take-an-Elevator Task

The *take-an-elevator* task is decomposed into the following steps:

(1) Move to the elevator hall from the current position. This step can be achieved by the free space recognition and the motion planning ability of the robot [10], provided that the route to the elevator hall is given.
(2) Move to the place in front of the button outside the elevator, where the manipulator can reach the button. The robot recognizes the elevator and localizes itself with respect to the elevator's local coordinates. For the movement, the robot sets a trajectory from the current position to the target position, and follows it by a sensory-feedback control.
(3) Localize the button and push it using the manipulator. The robot detects that the button is pushed by recognizing that the light of the button turns on.
(4) Move to the position in front of the elevator door where the robot waits for the door to open.
(5) Get on the elevator after recognizing the door's opening.
(6) Localize and push the button of the destination floor inside the elevator, as the same as (3).
(7) Get off the elevator after recognizing that the door opens (currently, the arrival at the target floor is not verified using floor signs inside the elevator).
(8) Move to the destination position at the target destination floor, as the same as (1).

Based on this analysis, we developed the task model for the *take-an-elevator* task. Fig. 4 shows that the robot can take an elevator autonomously by following the task model.

4 Teaching Examples

The robot examines the task model, and if there are missing pieces of knowledge in it, the robot acquires them through the interaction with the user. Each missing piece of knowledge needs the corresponding teaching procedure.

The above steps of the *take-an-elevator* task are divided into the following two parts. Steps (1) and (8) are composed of free movements. The other steps are composed of guarded movements near the elevator and hand motions. The following two subsections explain the teaching methods for the first and the second parts, respectively.

4.1 Route Teaching

The robot needs a free space map and a destination or a route to perform a *free movement*. The free space map is generated by the map generation capability of the robot, which is already embedded [8]. The destination may be given by some coordinate values, but they are not intuitive for the user to teach. So we take the following "teaching by guiding" approach [4, 5].

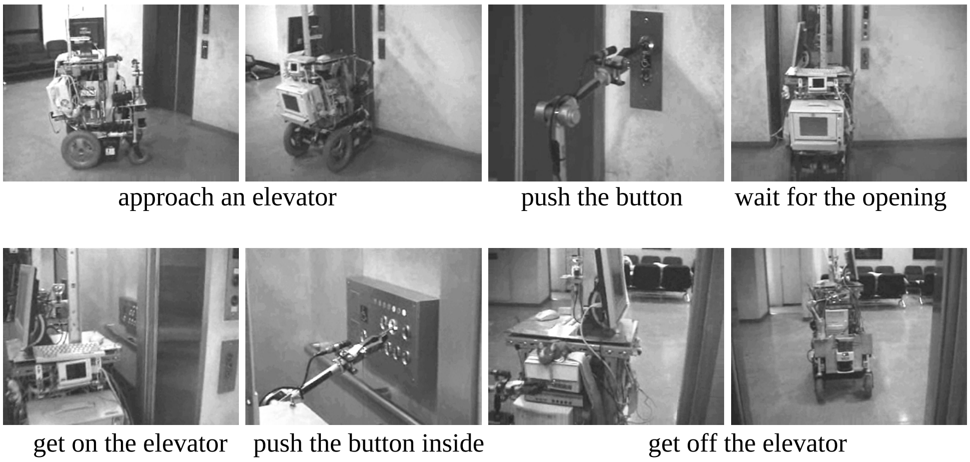


Fig. 4: The mobile robot is taking an elevator.

In route teaching, we first take the robot to a destination. During this *guided* movement, the robot learns the route. Then the robot can reach the destination by localizing itself with respect to the learned route. Such two-phase methods have been developed for both indoor and outdoor mobile robots; some of them are map-based [5, 6] and some are view-based [4, 7].

In this work, the robot simply memorizes the trace of its guided movement. Although the estimated trace suffers from accumulated errors, the robot can safely follow the learned route because of the reliable map generation; the robot moves to the direction of the destination within the recognized free space.

The next problem is how to guide the robot. In [4, 5], we used a joystick to control the robot; but this requires the user to know the mechanism of the robot. A user-friendly way is to implement a person-following function to the robot [2, 12]. For a simple and reliable person detection, we use a teaching device which has red LEDs; the user shows the device to the robot while he/she guides it to the destination (see Fig. 5). The robot repeatedly detects the device in both of the two omnidirectional camera by using a simple color-based detection algorithm, and calculates its relative position in the robot coordinates. The calculated position is input to our path planning method [10] as a temporary destination. Fig. 6 shows a snapshot of person tracking during a guided movement.

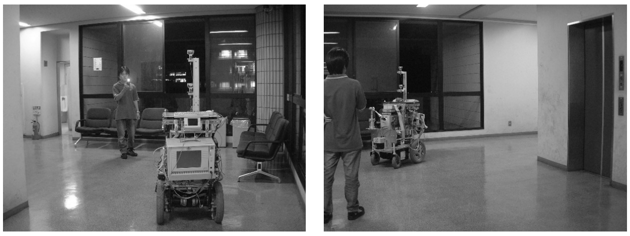


Fig. 5: Taking the robot to the destination.

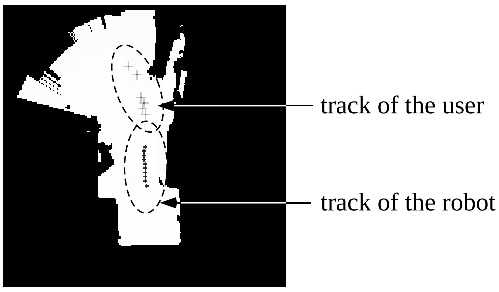


Fig. 6: Tracking the user. The white area is the detected free space.

4.2 Teaching of Vision-Based Operation

This section describes the methods for teaching the position of an elevator, the positions of buttons, and the views of them.

4.2.1 Teaching the Elevator Position

Suppose that the robot has already be taken to the elevator hall, using the method described above. The robot then asks about the position of the elevator. The user indicates it by pointing the door of the elevator (see Fig. 7). The robot has a general model of elevator shape, which is mainly composed of two parallel lines corresponding to the wall and the elevator door projected onto the floor. Using this model and the LRF (laser range finder) data, the robot searches the indicated area for the elevator and sets the origin of the elevator local coordinates at the center of the gap of the wall in front of the door (see Fig. 8).

4.2.2 Teaching the Button Position

The robot then asks where the buttons are, and the user indicates their rough position. The robot searches the indicated area on the wall for image patterns which match the given button models (e.g., circular or rectangular). Fig. 9 shows an example of detected button. The position of the button with respect to the elevator coordinates and the button view, which is used as an image template, are recorded after the verification by the user. The robot learns the buttons inside the elevator in a similar way; the user indicates the position of the button box, and the robot searches there for buttons.

Fig. 7: Teaching the elevator position to the robot.

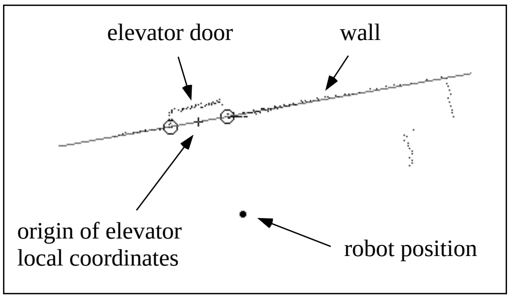


Fig. 8: Elevator detection from the LRF data.

Fig. 9: A detected button outside the elevator.

5 Conclusion

This paper has described a method of interactively teaching the task of taking elevators to a mobile robot. The method uses *task models* for describing the necessary pieces of knowledge for each task and their dependencies. Task models include the following three kinds of robot-specific knowledge: object models, motion models, and sensing skills. Using the task model, the robot can determine what pieces of knowledge are further needed, and plans necessary interactions with users to obtaining them. By this method, the user can teach only the important pieces of task knowledge easily and efficiently. We have shown the preliminary implementation and experimental results on the *take-an-elevator* task.

Currently the task model is manually designed for the specific, *take-an-elevator* task from scratch. It would be desirable, however, that a part of existing task models can be reused for describing another. Since reusable parts are in general commonly-used, typical operations, a future work is to develop a repertoire of typical operations by, for example, using an inductive learning-based approach [1, 13]. By using the repertoire, the user's effort for task modeling is expected to be reduced drastically.

Another issue is the development of teaching procedures. Although the mechanism of determining missing pieces of knowledge in a dependency network is general, for each missing piece, the corresponding procedure for obtaining it from the user should be provided. Such teaching procedures are also designed manually at present and, therefore, the kinds of pieces of knowledge that can be taught are limited. Implementing the procedures for various pieces of knowledge requires much user's effort, especially for non-symbolic (e.g., geometric or photometric) knowledge. Another future work is thus to develop interfaces that can be used for teaching a variety of non-symbolic knowledge. Graphical user interfaces (GUIs) (e.g., [11]) or multi-modal interfaces (MMIs) (e.g., [3]) are suitable for this purpose.

References

[1] B. Dufay and J.C. Latombe. An Approach to Automatic Robot Programming Based on Inductive Learning. *Int. J. of Robotics Research*, Vol. 3, No. 4, pp. 3–20, 1984.

[2] E. Huber and D. Kortenkamp. Using Stereo Vision to Pursue Moving Agents with a Mobile Robot. In *Proceedings of 1995 IEEE Int. Conf. on Robotics and Automation*, pp. 2340–2346, 1995.

[3] S. Iba, C.J. Paredis, and P.K. Khosla. Interactive Multi-Modal Robot Programming. In *Proceedings of 2002 IEEE Int. Conf. on Robotics and Automation*, pp. 161–168, 2002.

[4] H. Katsura, J. Miura, M. Hild, and Y. Shirai. A View-Based Outdoor Navigation Using Object Recognition Robust to Changes of Weather and Seasons. In *Proceedings of 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2974–2979, 2003.

[5] K. Kidono, J. Miura, and Y. Shirai. Autonomous Visual Navigation of a Mobile Robot Using a Human-Guided Experience. *Robotics and Autonomous Systems*, Vol. 40, No. 2-3, pp. 121–130, 2002.

[6] S. Maeyama, A. Ohya, and S. Yuta. Autonomous Mobile Robot System for Long Distance Outdoor Navigation in University Campus. *J. of Robotics and Mechatronics*, Vol. 9, No. 5, pp. 348–353, 1997.

[7] Y. Matsumoto, M. Inaba, and H. Inoue. Visual Navigation using View-Sequenced Route Representation. In *Proceedings of 1996 IEEE Int. Conf. on Robotics and Automation*, pp. 83–88, 1996.

[8] J. Miura, Y. Negishi, and Y. Shirai. Mobile Robot Map Generation by Integrating Omnidirectional Stereo and Laser Range Finder. In *Proceedings of 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 250–255, 2002.

[9] J. Miura, Y. Yano, K. Iwase, and Y. Shirai. Task Model-Based Interactive Teaching. In *Proceedings of IROS2004 Workshop on Issues and Approaches on Task Level Control*, 2004.

[10] Y. Negishi, J. Miura, and Y. Shirai. Adaptive Robot Speed Control by Considering Map and Localization Uncertainty. In *Proceedings of the 8th Int. Conf. on Intelligent Autonomous Systems*, pp. 873–880, 2004.

[11] F. Saito and T. Suehiro. Toward Telemanipulation via 2-D Interface – Concept and First Result of Titi. In *Proceedings of IECON 02*, pp. 2243–2248, 2002.

[12] Y. Sawano, J. Miura, and Y. Shirai. Man Chasing Robot by an Environment Recognition Using Stereo Vision. In *Proceedings of the 2000 Int. Conf. on Machine Automation*, pp. 389–394, 2000.

[13] M. Tsuda, H. Ogata, and Y. Nanjo. Programming Groups of Local Models from Human Demonstration to Create a Model for Robotic Assmebly. In *Proceedings of 1998 IEEE Int. Conf. on Robotics and Automation*, pp. 530–537, 1998.